UNIVERSITY OF MICHIGAN

EECS 270: Intro to Logic Design

Midterm Exam 2

Prof. Karem A Sakallah

Wednesday November 17, 2021
6:00-8:00 p.m.

A - L: 1013 DOW ‖ M - RL: 1017 DOW ‖ S - T: 1006 DOW ‖ U - Z: 1018 DOW

Name: _____

UMID: _____

Honor Pledge:
"I have neither given nor received aid on this exam, nor have I concealed any violation of the Honor Code."

Signature: _____

Instructions:

- The exam is closed book except for **two** 8.5"x11" sheets of notes No electronics of any kind may be used.
- Print your name and student ID number and sign the honor pledge.
- Make sure your answers and meaningful work are on the pages with numbers at the bottom. We will be scanning and looking at only these pages: all work on the backs of pages will **not** be checked for determining partial credit.
- The exam consists of 12 problems with the point distribution indicated here. Please keep this in mind as you work through the exam. Use your time wisely.
- There are 13 pages in this exam. Make sure that you have all 13 pages and notify an instructor if you do not.

1.	/6
2.	/8
3.	/10
4.	/10
5.	/5
6.	/10
7.	/4
8.	/10
9.	/10
10.	/12
11.	/5
12.	/10
Total:	/100

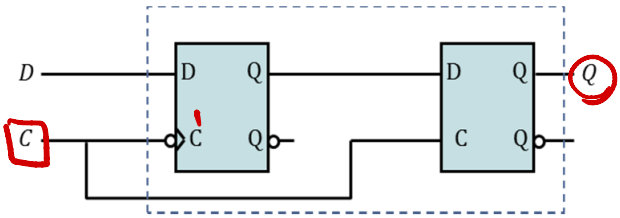
1 [Latch and and Flip-Flop Timing–6 points]

a. **[1 points]** If the propagation delay of the NOR gates in an SR latch is 5 ns, what is the *minimum allowable pulse width* on the S and R inputs to guarantee proper operation?

Answer: _____

b. **[1 points]**

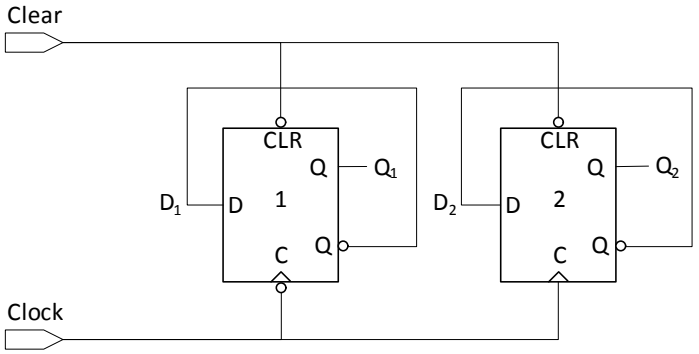
Consider the back-to-back negative edge-triggered flip-flop and positive level-sensitive latch shown here as a *single flip-flop*. Let $C\uparrow$ and $C\downarrow$ denote, respectively, the rising and falling edges of the clock input C, and let S and H denote the setup and hold times. Circle the correct answers below.

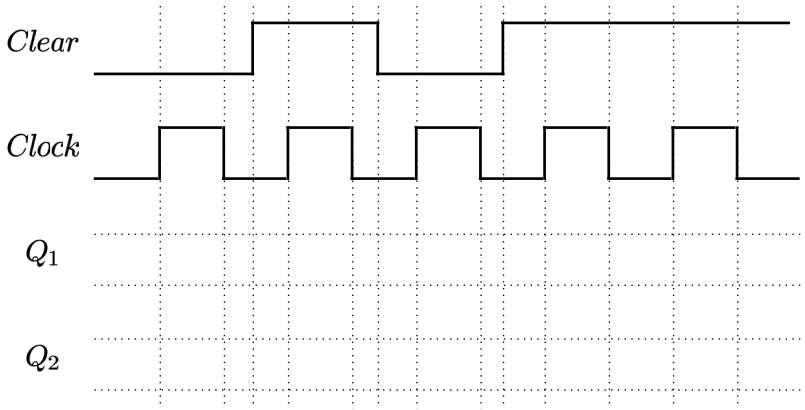


S and H must be checked against: $C\uparrow$ $C\downarrow$

Q changes shortly after: $C\uparrow$ $C\downarrow$

c. **[4 points]** The following circuit has two D flip-flops with asynchronous active-low "CLR" inputs. Complete the timing diagram for Q_1 and Q_2 assuming zero combinational delays. *You do not need to show causality arrows.*

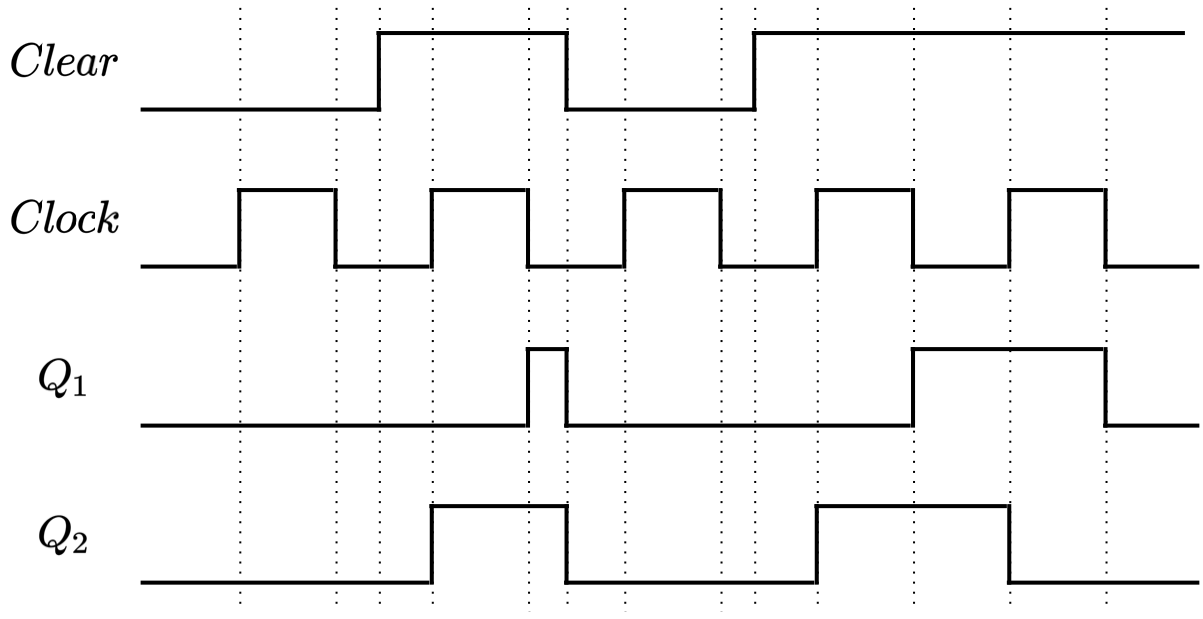




a. **[1 points]** 10 ns

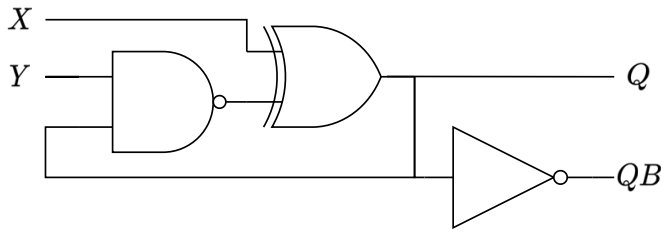
b. **[1 points]** S/H: $C\downarrow$, Q: $C\uparrow$

c. **[4 points]** Note that $Q_1^+ = D_1 = Q_1'$ and that $Q_2^+ = D_2 = Q_2'$



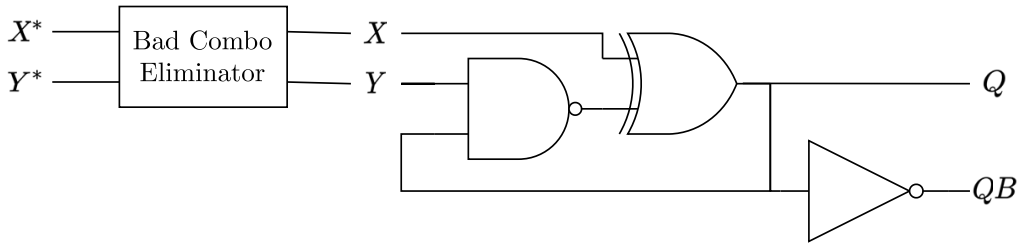
2 [Weird Latch–8 points]

a. **[4 points]** The XOR/NAND/INV circuit shown below is proposed as a new type of latch. Derive its transition table, indicating the function it performs (Set, Reset, Hold, Invalid). For invalid input combinations, state the reasons for their invalidity (e.g., Q and QB are not complements, Q and QB oscillate, etc.)



X	Y	Q^+	Function
0	0		
0	1		
1	0		
1	1		

b. **[4 points]** The invalid input combination(s) in this latch can be eliminated by "merging" them with one of the valid combinations (Set, Reset, or Hold). This can be accomplished by adding a "Bad Combo Eliminator" circuit as shown below. Design a minimal 2-input/2-output gate-level circuit (using the usual gates) that accepts all possible combinations on its X^*Y^* inputs, while producing only valid combinations on its XY outputs. Write the corresponding logic expressions for X and Y in terms of X^* and Y^*.



$X =$ _____

$Y =$ _____

[4 points]

X	Y	Q^+	Function
0	0	1	Set
0	1	Oscillation	Invalid
1	0	0	Reset
1	1	Q	Hold

[4 points] The truth table below shows the three possibilities of "eliminating" the invalid combination:

	Merge with Set	Merge with Reset	Merge with Hold
X^*Y^*	XY	XY	XY
0 0	0 0	0 0	0 0
0 1	0 0	1 0	1 1
1 0	1 0	1 0	1 0
1 1	1 1	1 1	1 1

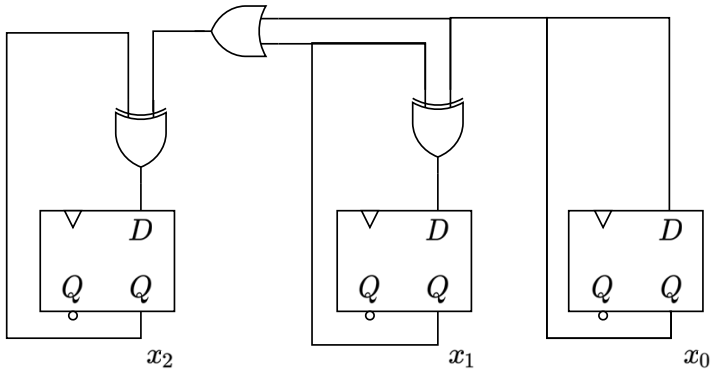
yielding the following solutions:

1. Merge with 00 (Set): $X = X^*$, $Y = X^*Y^*$. Cost: {AND2}.
2. Merge with 10 (Reset): $X = X^* + Y^*$, $Y = X^*Y^*$. Cost: {AND2, OR2}.
3. Merge with 11 (Hold): $X = X^* + Y^*$, $Y = Y^*$. Cost: {OR2}.

Thus, there are two minimal solutions each requiring a single 2-input gate.

3 [Sequential Circuit Analysis–10 points]

Let $X = x_2x_1x_0$ denote a two's complement number that has been loaded into the 3-bit register shown below. (The logic for loading the register is not shown.)



a. **[3 points]** Write the symbolic expressions of the next-state for each bit of X using the usual logic operators (AND, OR, XOR, etc.) You do not need to simplify the expressions.

$x_0^+ =$ _____

$x_1^+ =$ _____

$x_2^+ =$ _____

b. **[6 points]** Write the simplest possible symbolic expressions for each bit of X after *exactly two clock ticks. Hint:* $a + b = a \oplus b \oplus (ab)$.

$x_0^{++} =$ _____

$x_1^{++} =$ _____

$x_2^{++} =$ _____

c. **[1 points]** Assuming the register is initialized to $X = -3_{10}$, what is its value, in decimal notation, after two clock ticks?

Answer: $X =$ _____

[3 points]
$x_0^+ = x_0$
$x_1^+ = x_0 \oplus x_1$
$x_2^+ = (x_0 + x_1) \oplus x_2$

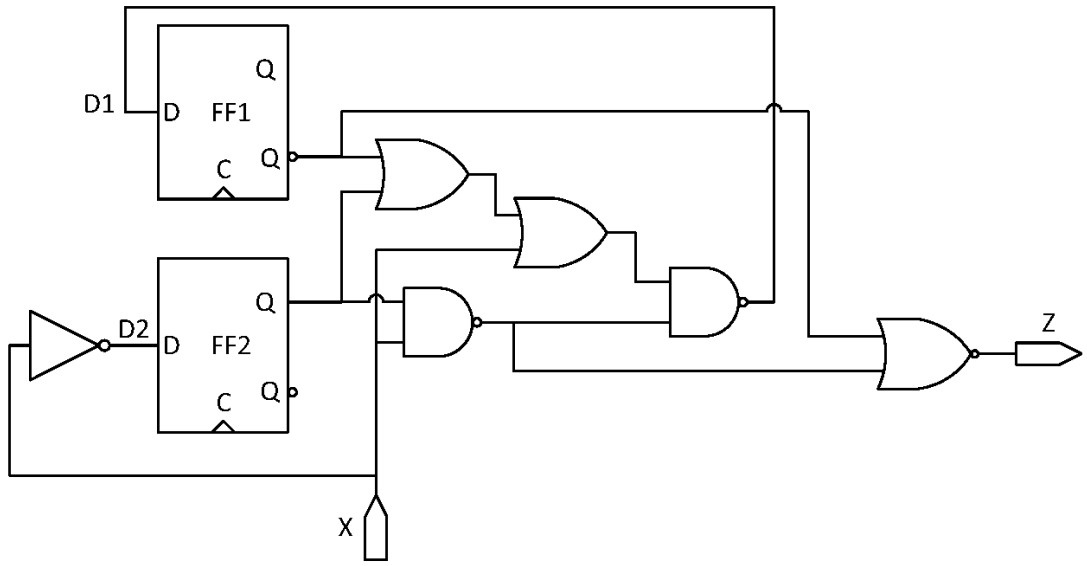
[6 points]

$$
\begin{aligned}
x_0^{++} &= x_0^{+} = x_0 \\
x_1^{++} &= x_0^{+} \oplus x_1^{+} \\
&= x_0 \oplus (x_0 \oplus x_1) \\
&= (x_0 \oplus x_0) \oplus x_1 \\
&= 0 \oplus x_1 = x_1 \\
x_2^{++} &= (x_0^{+} + x_1^{+}) \oplus x_2^{+} \\
&= (x_0 + (x_0 \oplus x_1)) \oplus ((x_0 + x_1) \oplus x_2) \\
&= x_0 \oplus (x_0 \oplus x_1) \oplus x_0(x_0 \oplus x_1) \oplus (x_0 \oplus x_1 \oplus x_0 x_1 \oplus x_2) \\
&= x_0 \oplus x_0 \oplus x_1 \oplus x_0 \oplus x_0 x_1 \oplus x_0 \oplus x_1 \oplus x_0 x_1 \oplus x_2 \\
&= x_2
\end{aligned}
$$

[1 points] Answer: -3_{10}

4 [Sequential Circuit Analysis–10 points]

This sequential circuit sets its output Z to 1 when the appropriate secret code is entered on its input X. Assuming that the circuit is reset to state $Q_1Q_2 = 00$ before application of any input, your job is to figure out what the secret code is. *Hint: Note that Z is a Mealy output!*



a. **[3 points]** Derive the next-state and output equations:

$Q_1^+ =$ _____

$Q_2^+ =$ _____

$Z =$ _____

b. **[4 points]** Complete the following transition/output table. Note that each table entry should be formatted as "Next State/Output".

Q_1Q_2	$X = 0$	$X = 1$
0 0		
0 1		
1 0		
1 1		

$Q_1^+Q_2^+/Z$

c. **[3 points]** What is the secret code?

Secret Code: _____

a. **[3 points]**

$Q_1^+ = [(XQ_2)'(X + Q_1' + Q_2]' = XQ_2 + X'Q_1Q_2'$
$Q_2^+ = X'$
$Z = [(Q_1' + (XQ_2)']' = XQ_1Q_2$

b. **[4 points]**

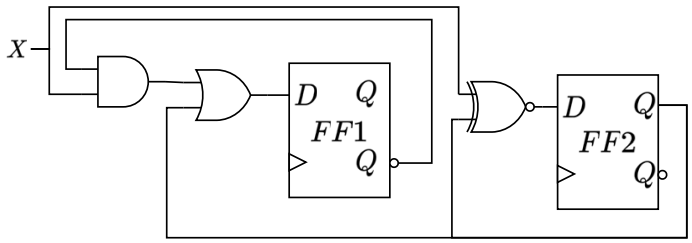
Q_1Q_2	$X = 0$	$X = 1$
0 0	0 1 / 0	0 0 / 0
0 1	0 1 / 0	1 0 / 0
1 0	1 1 / 0	0 0 / 0
1 1	0 1 / 0	1 0 / 1

$Q_1^+Q_2^+/Z$

c. **[3 points]** Secret Code: 0101.

5 [Sequential Circuit Analysis–5 points]

Consider the following synchronous sequential circuit and corresponding state assignment. Which of the following choices is its correct state table?



State Assignment*

Q_1Q_2	State Label
0 0	A
0 1	B
1 0	C
1 1	D

*Note that Q_1 is the most significant bit.

A.

PS	$X=0$	$X=1$
A	A	B
B	B	C
C	A	D
D	B	C

NS

B.

PS	$X=0$	$X=1$
A	B	C
B	C	D
C	B	A
D	C	D

NS

C.

PS	$X=0$	$X=1$
A	C	D
B	D	A
C	C	B
D	D	A

NS

D.

PS	$X=0$	$X=1$
A	D	A
B	A	B
C	D	C
D	A	B

NS

E.

PS	$X=0$	$X=1$
A	C	B
B	B	A
C	C	D
D	B	A

NS

$Q_1^+ = Q_2 + XQ_1'$
$Q_2^+ = Q_2 \odot X$

Q_1Q_2	$X=0$	$X=1$
0 0	0 1	1 0
0 1	1 0	1 1
1 0	0 1	0 0
1 1	1 0	1 1

$Q_1^+Q_2^+$

Answer: B

6 [Sequential Circuit Design–10 points]

Design a sequential two's complement circuit. The circuit has a single input x and a single output z. The input is assumed to be an arbitrarily long two's complement number $\cdots x_3x_2x_1x_0$ fed to the circuit serially starting with its least significant bit. The output z should be the two's complement negation of x.

Hint: $z_0 = x_0$, and for $i \geqslant 1$ $z_i = \begin{cases} x_i & \text{if } x_j = 0 \text{ for all } j < i \\ x_i' & \text{if } x_j = 1 \text{ for any } j < i \end{cases}$

a. **[2 points]** Write a symbolic expression for $z_i, i \geq 1$ in terms of $x_0, x_1, \cdots x_{i-1}, x_i$ using the least number of n-input logic operators from the set {AND, NAND, OR, NOR, XOR, XNOR} where $n \geq 2$ (i.e., you are not restricted to just 2-input operators).

$z_i =$ _____

b. **[4 points]** Derive suitable state/output and transition/output tables that implement your design.

c. **[4 points]** Draw the circuit that implements your design

[2 points] The hint suggests that z is a Mealy output which can be expressed as:

$$z_i = (x_0' x_1' \cdots x_{i-1}') \ ? \ x_i : x_i' = (x_0 + x_1 + \cdots + x_{i-1}) \ ? \ x_i' : x_i$$

Thus, other than z_0, each subsequent output bit z_i is equal to either x_i or x_i' depending on the truth value of $x_0 + x_1 + \cdots + x_{i-1}$. This can be implemented using a single XOR gate acting as a conditional inverter:

$$z_i = (x_0 + x_1 + \cdots + x_{i-1}) \oplus x_i$$

[4 points] Since the input bits are fed serially, we just need a single flip-flop that is initialized to 0 (to take care of $z_0 = x_0$) and that "remembers" if any previous input bit was 1 (to take care of all subsequent output bits).

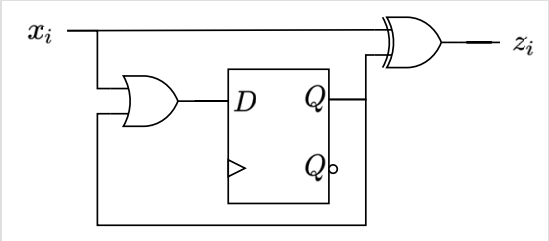
State Table: NS / z

PS	$x = 0$	$x = 1$
A	A / 0	B / 1
B	B / 1	B / 0

Transition Table: Q^+ / z

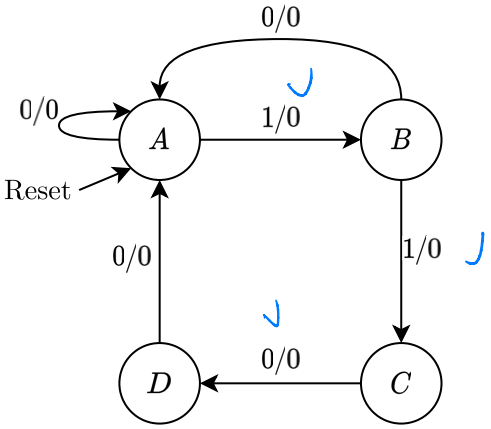
Q	$x = 0$	$x = 1$
0	0 / 0	1 / 1
1	1 / 1	1 / 0

[4 points]



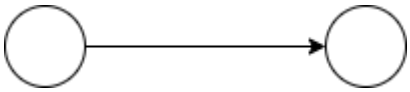
7 [Sequential Circuit Design–4 points]

A partially-completed state diagram for a sequence recognizer with input x and output z is shown below. The output should be 1 whenever the pattern 1101 including overlaps, appears on the input; otherwise it should be 0. Each arrow in the diagram is labeled with the value of x and corresponding value of z for that transition (e.g., the transition from A to B occurs when $x = 1$ and sets z to 0). Assume that A is the initial state.

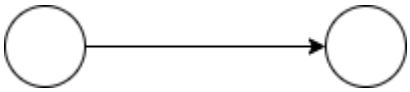


Identify the two missing transitions in this diagram.

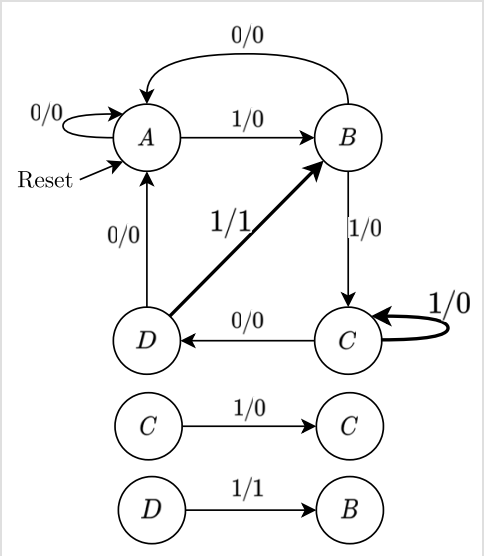
First missing transition:



Second missing transition:

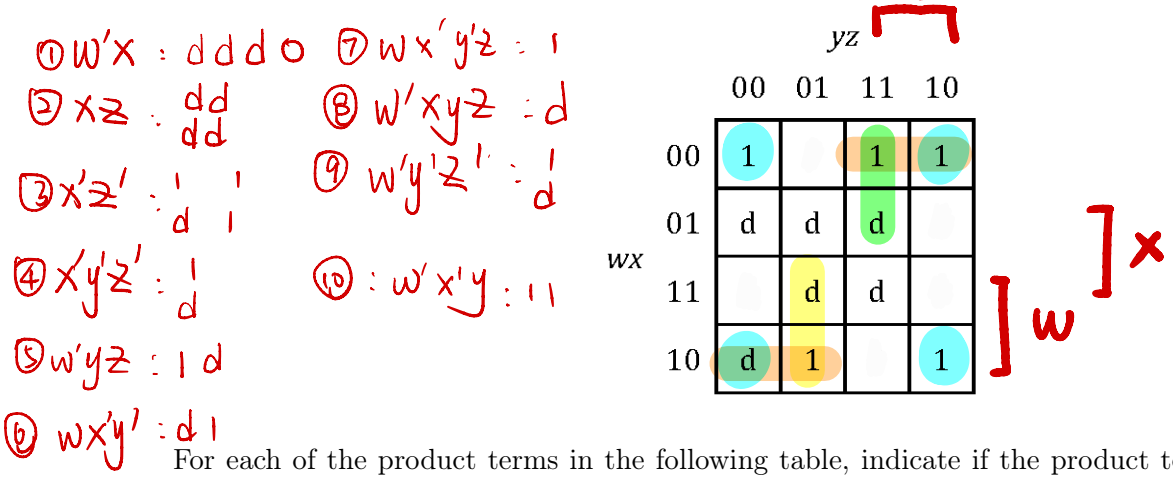


Missing transitions are **BOLDED**



8 [Two-Level Minimization–10 points]

Consider the following K-Map for the 4-variable function $f(w, x, y, z)$:



For each of the product terms in the following table, indicate if the product term is an implicant of f, and whether it is prime and/or essential. Indicate your answer by writing "Yes", "No", or "N/A" (for not applicable). N/A means that the product term a) is not an implicant of f, b) is an implicant that is not prime, or c) is a prime implicant that is not essential.

Hint: It helps if you first find ***all*** *the prime implicants before filling the table!*

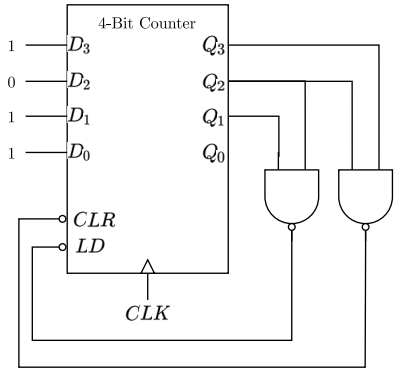
Product Term	$w'x$	xz	$x'z'$	$x'y'z'$	$w'yz$	$wx'y'$	$wx'y'z$	$w'xyz$	$w'y'z'$	$w'x'y$
Implicant?	No	No	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes
Prime Implicant?	N/A	N/A	Yes	No	Yes	Yes	No	N/A	Yes	Yes
Essential PI?	N/A	N/A	Yes	N/A	No	No	N/A	N/A	No	No

Product Term	$w'x$	xz	$x'z'$	$x'y'z'$	$w'yz$	$wx'y'$	$wx'y'z$	$w'xyz$	$w'y'z'$	$w'x'y$
Implicant?	No	No	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes
Prime Implicant?	N/A	N/A	Yes	No	Yes	Yes	No	N/A	Yes	Yes
Essential PI?	N/A	N/A	Yes	N/A	No	No	N/A	N/A	No	No

9 [Sequential Blocks–10 points]

a. **[3 points]**

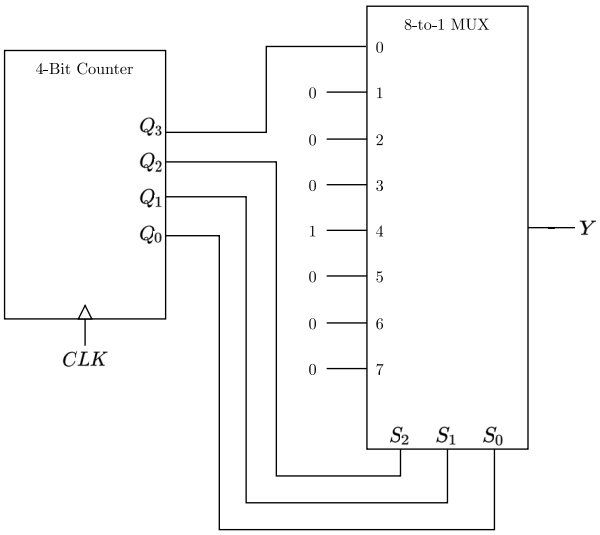
This circuit uses a binary counter that has synchronous load and clear controls. What is its modulus?



Answer: _____

b. **[3 points]**

A 4-bit binary up-counter is connected to an 8-to-1 MUX. As the counter cycles through its 16 states (from 0 to 15), the output Y will be repeatedly asserted (becomes 1) at a subset of these states. Determine this subset as an increasing sequence of decimal values.



Answer: _____

c. **[4 points]**

An 8-bit Johnson counter is constructed from a left-shift register $Q_7Q_6 \cdots Q_0$. The next-state equations for each of the register's 8 flip-flops are:

$Q_0^+ =$ _____ $Q_4^+ =$ _____

$Q_1^+ =$ _____ $Q_5^+ =$ _____

$Q_2^+ =$ _____ $Q_6^+ =$ _____

$Q_3^+ =$ _____ $Q_7^+ =$ _____

a. **[3 points]** The counter is loaded with $1011_2 = 11_{10}$ when the count reaches 6 and is cleared when the count reaches 12. The counting sequence is 0, 1, 2, 3, 4, 5, 6, 11, 12 giving a modulus of 9.

b. **[3 points]**

$Q_2Q_1Q_0$	$Q_3 = 0$	$Q_3 = 1$
0	0	1
1	0	0
2	0	0
3	0	0
4	1	1
5	0	0
6	0	0
7	0	0

$Y = 1$ when $Q_3Q_2Q_1Q_0 = -100$ and when $Q_3Q_2Q_1Q_0 = 1000$ Thus, Y is asserted at counts 4, 8, and 12.

c. **[3 points]**

$$Q_0^+ = Q_7' \quad Q_4^+ = Q_3$$
$$Q_1^+ = Q_0 \quad Q_5^+ = Q_4$$
$$Q_2^+ = Q_1 \quad Q_6^+ = Q_5$$
$$Q_3^+ = Q_2 \quad Q_7^+ = Q_6$$

10 [Sequential Timing Analysis–12 points]

The following tables show the minimum and maximum combinational delays, as well as the timing parameters of the flip-flops, in a sequential circuit with three positive edge-triggered D flip-flops. All delays and timing parameters are in nanoseconds.

Minimum Delays

δ_{ij}	D_0	D_1	D_2
Q_0	1	2	3
Q_1	2	1	∞
Q_2	∞	0	2

Maximum Delays

Δ_{ij}	D_0	D_1	D_2
Q_0	5	3	7
Q_1	3	4	$-\infty$
Q_2	$-\infty$	4	5

FF Timing Parameters

Clock to Q delay:	2
Setup time:	2
Hold time:	1

The timing specification for this circuit requires it to operate at a frequency of 100MHz.

a. **[6 points]** Compute the early and late arrival times for the three flip-flops.

$a_0 =$ _____ $A_0 =$ _____

$a_1 =$ _____ $A_1 =$ _____

$a_2 =$ _____ $A_2 =$ _____

b. **[3 points]** Determine if there are any setup or hold violations:

Hold violation at FF0	True	False
Hold violation at FF1	True	False
Hold violation at FF2	True	False
Setup violation at FF0	True	False
Setup violation at FF1	True	False
Setup violation at FF2	True	False

c. **[3 points]** If there are setup or hold violations that prevent the circuit from operating at 100MHz, what is the minimum clock period P_{min} in nanoseconds and corresponding clock frequency f_{max} in MHz that allow the circuit to function without timing errors? *Note: Truncate f_{max} to one digit after the decimal point.*

$P_{\text{min}} =$ _____

$f_{\text{max}} =$ _____

a. **[6 points]**

$a_0 = \min(2+1, 2+2) = 3$ ns
$a_1 = \min(2+2, 2+1, 2+0) = 2$ ns
$a_2 = \min(2+3, 2+2) = 4$ ns

$A_0 = \max(2+5, 2+3) = 7$ ns
$A_1 = \max(2+3, 2+4, 2+4) = 6$ ns
$A_3 = \max(2+7, 2+5) = 9$ ns

b. **[3 points]**

Hold violation at FF0	True	[False]
Hold violation at FF1	True	[False]
Hold violation at FF2	True	[False]
Setup violation at FF0	True	[False]
Setup violation at FF1	True	[False]
Setup violation at FF2	[True]	False

c. **[3 points]**

$P_{\min} = \max(2+7, 2+6, 2+9) = 11$ ns

$f_{\max} = \frac{1}{11 \times 10^{-9}} = \frac{1000 \times 10^6}{11} = 90.9$ MHz

11 [Potpourri–5 points]

A selection of multiple-choice and True/False questions on various topics!

a. **[1 points]**

The main difference between Moore and Mealy outputs is:

A. Moore outputs are synchronous with the clock whereas Mealy outputs are asynchronous with the clock
B. Moore outputs are functions of the inputs only whereas Mealy outputs are functions of both the inputs and the state variables
C. Moore outputs are functions of the state variables only whereas Mealy outputs are functions of both the state variables and the inputs
D. Moore outputs are functions of primary inputs and state variables whereas Mealy outputs are functions of state variables only.
E. None of the above

b. **[1 points]**

An example of an asynchronous counter is:

A. A decade counter
B. A Johnson counter
C. A ring counter
D. A ripple binary counter
E. None; All are synchronous

c. **[1 points]**

A minimal SOP solution for a Boolean function must include at least one essential prime implicant.

A. TRUE
B. FALSE

d. **[1 points]**

The counting sequence, in decimal, of a 2-bit Johnson counter is: $0 \rightarrow$ _____

e. **[1 points]**

The number of prime implicants of an $n-$input OR function is _____

a. **[1 points]** C

b. **[1 points]** D

c. **[1 points]** FALSE

d. **[1 points]** $0 \rightarrow 1 \rightarrow 3 \rightarrow 2$

e. **[1 points]** n

12 [Lab Experience–10 points]

Complete the following Verilog that implements a counter that counts 0,1,2,3,2,1 and repeats.

```
module counter(input reset, clk, output reg [1:0] state);

  BLANK1 NS1, NS0;     // next state
  BLANK2 S1, S0;       // current state
  reg D;               // direction: 1 is up, 0 is down

  assign NS1 = ~D & S1 & S0 | BLANK3 | D & S1 & ~S0;
  assign NS0 = BLANK4;

  BLANK5
  begin
    if( reset ) begin S1 <= 0; S0 <= 0 ; D <= 1; end
    else begin BLANK6; end
    if (BLANK7) D <= 0; else BLANK8;
  end

  BLANK9
  begin
    state [1] = S1;
    state [0] = S0;
  end
BLANK10
```

BLANK1: _____ BLANK6 : _____

BLANK2: _____ BLANK7 : _____

BLANK3: _____ BLANK8 : _____

BLANK4: _____ BLANK9 : _____

BLANK5: _____ BLANK10 : _____

```
module counter(input reset, clk, output reg [1:0] state);

  wire NS1, NS0;     // next state
  reg S1, S0;       // current state
  reg D;              // direction: 1 is up, 0 is down

  assign NS1 = ~D & S1 & S0 | D & -S1 & S0 | D & S1 & -S0;
  assign NS0 = -S0;

  always @(posedge clk)
  begin
    if( reset ) begin S1 <= 0; S0 <= 0 ; D <= 1; end
    else begin S1 <= NS1; S0 <= NS0; end
    if (S1) D <= 0; else D <= 1;
  end

  always @*
  begin
    state [1] = S1;
    state [0] = S0;
  end
endmodule
```

BLANK1:	wire	BLANK6 :	S1 <= NS1; S0 <= NS0
BLANK2:	reg	BLANK7 :	S1
BLANK3:	D & -S1 & S0	BLANK8 :	D <= 1
BLANK4:	-S0	BLANK9 :	always @*
BLANK5:	always @(posedge clk)	BLANK10 :	endmodule